(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 January 2004 (08.01.2004)

PCT

(10) International Publication Number
WO 2004/004198 A1

(51) International Patent Classification[7]: H04L 9/16, 9/08, H04N 1/41, 1/44

(21) International Application Number: PCT/JP2003/007976

(22) International Filing Date: 24 June 2003 (24.06.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2002-191284 28 June 2002 (28.06.2002) JP

(71) Applicant *(for all designated States except US)*: CANON KABUSHIKI KAISHA [JP/JP]; 3-30-2, Shimomaruko, Ohta-ku, Tokyo 146-8501 (JP).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: HAYASHI, Junichi [JP/JP]; c/o CANON KABUSHIKI KAISHA, 3-30-2, Shimomaruko, Ohta-ku, Tokyo 146-8501 (JP).

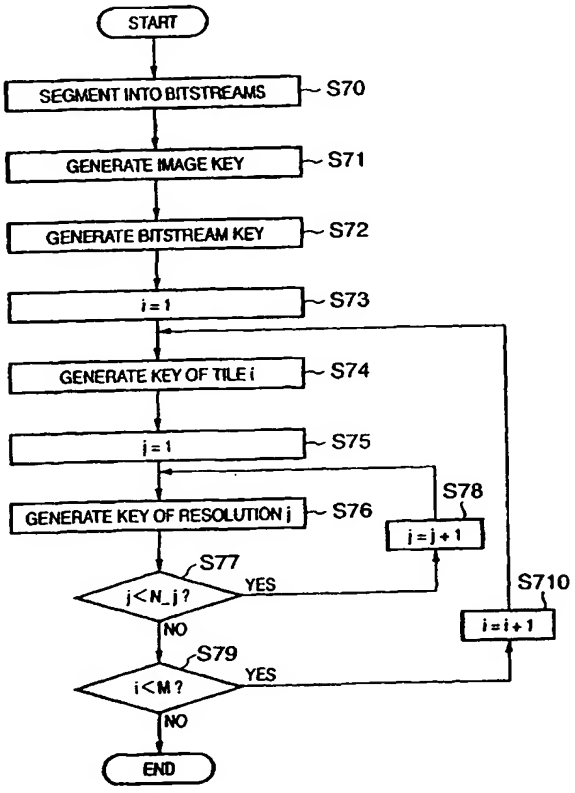
(74) Agent: OHTSUKA, Yasunori; 7th FL., SHUWA KIOICHO PARK BLDG., 3-6, KIOICHO, CHIYODA-KU, Tokyo 102-0094 (JP).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,

[Continued on next page]

(54) Title: INFORMATION PROCESSING METHOD, INFORMATION PROCESSING APPARATUS, PROGRAM, AND STORAGE MEDIUM



(57) Abstract: A bitstream of each tile in a code sequence is segmented into bitstreams for respective resolution levels (S70). Image key K I is generated (S71). Bitstream key K BS is generated from image key K I using encoding parameters P (S72). Key K Ti of tile Ti is generated from bitstream key K BS using information associated with that tile (S74). Finally, keys corresponding to the segmented bitstreams indicating respective resolutions in tile Ti are generated (S76).

WO 2004/004198 A1

DESCRIPTION

INFORMATION PROCESSING METHOD, INFORMATION PROCESSING APPARATUS, PROGRAM, AND STORAGE MEDIUM

TECHNICAL FIELD

The present invention relates to an information processing apparatus for encrypting an image, an information processing method and information processing apparatus for decrypting an encrypted image, a program, and a storage medium.

BACKGROUND ART

Conventionally, entire image data undergoes encryption, scrambling, or the like to transfer image data or the like in secrecy. This is a technique for encrypting entire image data using an encryption key, and allowing only a party who has a decryption key corresponding to the encryption key to normally decrypt the encrypted image data.

Especially, image data having a hierarchical structure undergoes an encryption process using different encryption keys for respective layers for the purpose of controlling reproduction of the image data in correspondence with the hierarchical structure. Also, image data made up of a plurality of tiles undergoes an encryption process using different encryption keys for respective tiles for the purpose of controlling

image, the low-resolution components are decrypted using the key corresponding to the low-resolution components, and the high-resolution components remain encrypted. In such case, image data obtained by multiplexing noise data of the high-resolution component on the image data of the low-resolution components is reproduced, and the low-resolution image cannot be browsed without being multiplexed with the noise data.

When image data is encrypted using different encryption keys for respective tiles and layers, there is no means for determining correspondence between encrypted predetermined tiles and layers, and decryption keys required to decrypt them. For this reason, a decryption process cannot often be normally made. Likewise, when some tiles or layers of image data are encrypted, it is difficult to discriminate encrypted tiles and layers from those which are not encrypted, and a decryption process cannot often be normally made.

The present invention has been made in consideration of the above problems, and has as its object to allow easy management of keys used in encryption.

DISCLOSURE OF INVENTION

In order to achieve the above object, for example, an information processing method of the present invention comprises the following arrangement.

first recognition means for recognizing a first unit n (n = 1 to N) which specifies segmentation of encoded image data;

second recognition means for recognizing a second unit m (m = 1 to M) which specifies segmentation of encoded image data;

first parameter generation means for generating a first parameter X_n (n = 1 to N) on the basis of the first unit n; and

key parameter generation means for generating a key parameter corresponding to each combination (n, m) of the first and second units on the basis of at least the first parameter, and

in that the key parameter is used to encrypt or decrypt partial encoded image data D(n, m) corresponding to the combination (n, m) in the encoded image data, and

the first parameter generation means generates each first parameter X_n on the basis of an algorithm which uniquely determines the first parameter X_n based on a neighboring first parameter X_{n-1}.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF DRAWINGS

Fig. 6 is a flow chart of a key generation process executed by a key generation unit 12 according to the first embodiment of the present invention;

Fig. 7 shows an example of three tiles (T01, T02, and T03) which are respectively segmented into different numbers of bitstreams;

Fig. 8 is a view for explaining an example of a key ID generation method;

Fig. 9 shows an example of keys and key IDs corresponding to respective tiles and respective resolution levels of each tile when an image is made up of four tiles, tile 01 is made up of three resolution levels, tile 02 is made up of two resolution levels, tile 03 is made up of three resolution levels, and tile 04 is made up of four resolution levels;

Fig. 10 shows an example of the format of an ID reference table according to the first embodiment of the present invention;

Fig. 11 is a block diagram showing the functional arrangement of a decryption apparatus (decryption system) according to the first embodiment of the present invention;

Fig. 12 is a block diagram showing the functional arrangement of a second key generation unit 123 according to the first embodiment of the present invention;

Fig. 13 is a flow chart of a key generation process executed by the key generation unit 12 according to the second embodiment of the present invention;

Fig. 21 is a view showing the flow of sequentially decrypting a given region of a subband to be decrypted for respective bit planes and finally reconstructing quantization indices;

Fig. 22A is a block diagram showing the arrangement and process of an inverse discrete wavelet transformer 214;

Fig. 22B is a block diagram showing the arrangement and process of an inverse discrete wavelet transformer 214;

Fig. 23A shows an example of a code sequence;

Fig. 23B shows correspondence between respective subbands and the sizes of tiles to be displayed;

Fig. 24 shows an example wherein an image is made up of four tiles, and each tile has three resolution levels; and

Fig. 25 shows the format of a code sequence in which bit planes are arranged in the order from an upper bit plane to a lower bit plane, and each bit plane contains bitstreams of respective subbands.

BEST MODE FOR CARRYING OUT THE INVENTION

Preferred embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

[First Embodiment]

An encryption apparatus according to this embodiment for encrypting an image, and a decryption apparatus for

encoding unit 11 according to this embodiment comprises an image input unit 21, tile segmentation unit 22, discrete wavelet transformer 23, quantizer 24, entropy encoder 25, and code output unit 26.

Pixel signals which form an image to be encoded are input to the image input unit 21 in the raster scan order, and the output from the image input unit 21 is input to the tile segmentation unit 22. In the following description, an image signal expresses a monochrome multi-valued image. However, when a plurality of color components of, e.g., a color image or the like are to be encoded, each of R, G, and B color components or each of luminance and chromaticity components may be compressed as the monochrome component.

The tile segmentation unit 22 segments the input image signal into at least one tile having a predetermined size, and outputs the segmented tile or tiles. Note that the tiles indicate rectangular regions which do not overlap each other, and the subsequent processes are independently executed for each tile.

The discrete wavelet transformer 23 executes a two-dimensional wavelet transformation process for the input image signal, and computes and outputs transform coefficients. Fig. 3A shows the basic arrangement of the discrete wavelet transformer 23. An input image signal is stored in a memory 31, is sequentially read out by a processor 32 to undergo the transformation process, and is written in the memory 31 again. In this embodiment, Fig. 3B

The quantizer 24 quantizes the input coefficients by a predetermined quantization coefficient, and outputs indices corresponding to the quantized values. In this case, quantization is described by:

$$q = \mathrm{sign}(c)\ \mathrm{floor}(\mathrm{abs}(c)/\Delta) \quad (3)$$

$$\mathrm{sign}(c) = 1;\ c \geq 0 \quad (4)$$

$$\mathrm{sign}(c) = -1;\ c < 0 \quad (5)$$

where c is a coefficient to be quantized. In this embodiment, the value Δ includes "1". When $\Delta = 1$, no quantization is done in practice, and subbands input to the quantizer 24 are directly output to the entropy encoder 25.

The entropy encoder 25 decomposes the input quantization indices into bit planes, executes binary arithmetic coding for respective bit planes, and outputs a code stream.

Fig. 4 is a view for explaining the operation of the entropy encoder 25. In this example, a 4 × 4 code block region includes three nonzero quantization indices, which respectively have values "+13", "-6", and "+3". The entropy encoder 25 scans this region to obtain a maximum value M, and computes the number S of bits required for expressing the maximum quantization index by:

$$S = \mathrm{ceil}(\log 2(\mathrm{abs}(M))) \quad (6)$$

where ceil(x) is the smallest one of integers equal to or larger than x. In Fig. 4, since the maximum value is "13", S = 4. Hence, 16 quantization indices in the sequence are processed for respective four bit planes, as shown in the

the like) to the key generation unit 12. P represents the encoding parameters input to the key generation unit 12.

Figs. 5A, 5B, 5C, and 5D show the format of the code sequence which is generated and output in this way. Fig. 5A shows the overall format of the code sequence, in which MH is a main header; TH, a tile header; and BS, a bitstream. As shown in Fig. 5B, main header MH is comprised of the size (the numbers of pixels in the horizontal and vertical directions) of an image to be encoded, the size of each tile upon breaking up the image into tiles as a plurality of rectangular regions, the number of components indicating the number of color components, the size of each component, and component information indicating bit precision.

Fig. 5C shows the format of tile header TH. Tile header TH consists of a tile length including the bitstream length and header length of the tile of interest, and an encoding parameter for the tile of interest. The encoding parameter includes a discrete wavelet transformation level, filter type, and the like. Fig. 5D shows the format of bitstream BS of each tile. In Fig. 5D, a bitstream is formed for respective subbands, which are arranged in turn from a subband having a low resolution in ascending order of resolution. Furthermore, in each subband, codes are set for respective bit planes, i.e., in the order from an upper bit plane (MSB) to a lower bit plane (LSB). With this code arrangement, hierarchical decoding corresponding to

contains those of HL2, LH2, and HH2 subbands, and bitstream BS01 contains those of HL1, LH1, and HH1 subbands).

Therefore, when BS03 is decoded, a low-resolution image can be reproduced. When BS02 is decoded in turn, an image having a higher resolution can be reproduced. When BS01 is decoded finally, an image having the highest resolution can be reproduced. Note that a numeral that follows BS indicates a resolution index.

In Fig. 7, reference numeral 82 denotes an example in which the bitstream which forms tile T02 is segmented into three bitstreams as in T01 but another segmentation method is adopted. Furthermore, reference numeral 83 in Fig. 7 denotes an example in which tile T03 is segmented into two bitstreams. As described above, different bitstream segmentation methods can be independently set for respective tiles. The aforementioned segmentation process may be explicitly designated by the user or may be automatically designated using the number of bitstreams to be segmented, which is determined in advance.

The description will revert to Fig. 6. Upon completion of the bitstream segmentation process of respective tiles, image key K_I is generated (step S71). Image key K_I is the only one key for one image. Hence, in this embodiment, image key K_I is a key for the image to be encoded input to the image input unit 21. The generated image key K_I is saved together with a key ID which is also generated in step S71. The key ID is an index

Variable j is initialized to 1 (step S75). Variable j is a resolution index indicating each segmented bitstream described above. The resolution is higher with decreasing j, and vice versa.

In tile Ti, a key for a segmented bitstream indicating each resolution is generated (step S76). More specifically, key K_TiSj of resolution j is generated from key K_Ti of tile Ti. For example, key K_TiSj is generated using:

$$\text{If } j = 1,\ K\_TiSj = K\_Ti$$
$$\text{If } j \neq 1,\ K\_TiSj = H(K\_TiSj - 1) \quad (9)$$

The generated resolution key K_TiSj is saved together with a key ID which is also generated in step S76.

It is then checked if variable j < variable N_i (step S77). Note that parameter N_i is the number of segmented bitstreams which form tile Ti. That is, it is determined whether or not resolution keys K_TiSj corresponding to all (segmented) bitstreams which form tile Ti have been generated. If the checking result is true, the flow advances to step S78 to increment variable j by 1. Then, the aforementioned process in step S76 is executed using incremented variable j. On the other hand, if the checking result in step S77 is false, the flow advances to step S79 to check if variable i < parameter M. Note that parameter M indicates the total number of tiles. That is, it is determined whether or not tile keys K_Ti corresponding to all the tiles have been generated. If the checking result

three resolution levels, tile 02 is made up of two resolution levels, tile 03 is made up of three resolution levels, and tile 04 is made up of four resolution levels.

In step S71 above, image key K_IMG and its key ID "0000" are generated. In step S72 above, bitstream key K_BS and its key ID "0001" are generated. In step S74, tile key K_T1 and its key ID "0100", tile key K_T2 and its key ID "0200", tile key K_T3 and its key ID "0300", and tile key K_T4 and its key ID "0400" are generated. In step S76, resolution keys and their key IDs for respective resolutions of each tile are generated. Taking tile 1 as an example, resolution key K1S1 and its key ID "0101", resolution key K1S2 and its key ID "0102", and resolution key K1S3 and its key ID "0103" are generated.

In practice, since each tile key is equal to a resolution key of the highest resolution in each tile (for example, in case of tile 1, K_T1 and K_T1S1), the tile key or the resolution key of the highest resolution may be omitted. Furthermore, in this embodiment, the keys and key IDs have been independently explained for the sake of simplicity. However, in practice, pairs of keys and key IDs in this embodiment can be handled as keys required for decryption.

The aforementioned key ID is used to specify a tile and resolution level encrypted by the key corresponding to it. However, the location of the corresponding tile and resolution level in a bitstream cannot be determined from

The ID reference table generated using the aforementioned process must be transmitted to a decryption apparatus (to be described later) by a secure method. For example, the ID reference table may be encrypted or signed, and may be embedded in a code sequence. This is because if the ID reference table has been tampered with, a decryption process cannot be normally executed.

Furthermore, all the keys and key IDs shown in Fig. 9 need not always be saved and securely transmitted to the decryption apparatus. That is, of the keys shown in Fig. 9, only image key K_IMG is securely saved. Upon decryption, a key corresponding to a bitstream which is permitted to be decrypted may be generated from image key K_IMG, and only the generated key may be securely transmitted. In this way, only one key need be securely saved and managed for each image data, and such method is effective in terms of management and efficiency. Details of generation of a key required for decryption will be described later.

The process to be executed by the code sequence encryption processing unit 13 will be described below. The code sequence encryption processing unit 13 receives the code sequence, keys generated by the previous key generation process, and ID reference table. The unit 13 executes an encryption process of the code sequence using the keys and ID reference table, and outputs an encrypted code sequence. More specifically, the location of a bitstream to be encrypted in the code sequence is specified

unit 13 may be integrated. In this case, the encryption process is executed every time each resolution key is generated. That is, the encryption process may be executed simultaneously with generation of a resolution key in step S76.

<Decryption Apparatus (Decryption System)>

Fig. 11 is a block diagram showing the functional arrangement of a decryption apparatus (decryption system) according to this embodiment. As shown in Fig. 11, the decryption apparatus according to this embodiment comprises a first key generation unit 121, network unit 122, second key generation unit 123, code sequence decryption unit 124, and image decoding unit 125. Note that the network unit 122 may be omitted. In such case, the decryption apparatus comprises the first key generation unit 121, second key generation unit 123, code sequence decryption unit 124, and image decoding unit 125. Note that the decryption process to be described below may be executed by a computer which loads software with this arrangement (computer program). In this case, Fig. 11 corresponds to a block diagram that shows the functional arrangement of a program that makes a computer execute the decryption process according to this embodiment.

The process to be executed by the first key generation unit 121 will be described first. The first key generation unit 121 receives image key K_IMG and encoding parameters P. The unit 121 generates and outputs a key and key ID

from the generated tile key, and only the generated resolution key and its key ID are output.

As described above, a key and key ID corresponding to a bitstream to be decrypted are generated using the first key generation unit 121, and the generated key and key ID must be securely transmitted to the subsequent second key generation unit 123 via the network unit 122 (directly to the second key generation unit 123 if the network unit 122 is omitted).

The process executed by the second key generation unit 123 will be described below. The second key generation unit 123 receives the key and key ID output from the aforementioned first key generation unit, generates a decryption key corresponding to a bitstream to be decrypted by the code sequence decryption unit 124 (to be described later), and outputs the generated decryption key. In some cases, a plurality of decryption keys may be generated. A second key generation process executed by the second key generation unit 123 is basically the same as the key generation process that has been explained using Fig. 6. However, the second key generation unit 123 generates another key and key ID to have the key and key ID input to it as a start point in place of generating them to have image key K_IMG as a start point unlike in the key generation process that has been explained using Fig. 6.

The second key generation process will be described in detail below using Fig. 12. Fig. 12 is a block diagram

resolutions lower than the resolution level of that key can be generated. A case wherein the key ID does not include "00" will be referred to as "case D" hereinafter.

The arithmetic unit 132 makes the arithmetic operation for the input key on the basis of the determination result, thereby generating keys that can be generated.

For example, in case A, keys are generated by the same process as that in step S72 and subsequent steps in the method shown in Fig. 6. In case B, keys are generated by the same process as that in step S73 and subsequent steps in the method shown in Fig. 6. In case C, keys are generated by the same process as that in steps S75 to S78 in the method shown in Fig. 6. In case D, after the initial value of j is set to be equal to the resolution index included in the input key ID in step S75, keys are then generated by the same process as that in steps S75 to S78.

As described above, the second key generation unit determines keys that can be generated from the key input from the first key generation unit using the key ID input from the first key generation unit, and can generate keys and key IDs used in decryption by executing the arithmetic processes on the basis of the determination result.

The process executed by the code sequence decryption unit 124 will be described below. The code sequence decryption unit 124 receives keys generated by the second key generation unit, and the encrypted code sequence, and

The code input unit 211 receives the decrypted code sequence, analyzes the header included in that sequence to extract parameters required for the subsequent processes, and controls the flow of processes if necessary or outputs required parameters to the subsequent processing units. The bitstream included in the code sequence is output to the tile segmentation unit 216.

The tile segmentation unit 216 segments the input bitstream into those corresponding to tiles, and outputs the segmented bitstreams. The tile segmentation process is executed with reference to the main header, tile header, and the like. After the tile segmentation process, individual bitstreams are output to the entropy decoder 212. Note that the subsequent processes are independently executed for each bitstream obtained by the tile segmentation process.

The entropy decoder 212 decodes and outputs the bitstreams for respective bit planes. Fig. 21 shows the decoding sequence at that time. Fig. 21 illustrates the flow for sequentially decoding one subband region to be decoded for respective bit planes to finally restore a quantization index, and bit planes are decoded in the order of an arrow in Fig. 21. The restored quantization indices are output to dequantizer 213.

The dequantizer 213 reclaims discrete wavelet transform coefficients from the input quantization indices by:

Note that the forward and inverse discrete wavelet transformation processes given by equations (1), (2), (12), and (13) satisfy a perfect reconstruction condition. Hence, since the quantization step $\Delta = 1$ in this embodiment, the restored image signal x' matches an original image signal x if all bit planes are decoded in bit plane decoding.

With the aforementioned process, an image signal is reclaimed and is output to the image output unit 215. Note that the image output unit 215 may be an image display device such as a monitor or the like, or may be a storage device such as a magnetic disk or the like.

The image display pattern upon restoring and displaying an image in the aforementioned sequence will be explained using Figs. 23A and 23B. Fig. 23A shows an example of a code sequence, the basic format of which is based on Figs. 5A, 5B, 5C, and 5D. Since an image is made up of a plurality of tiles, the code sequence contains a plurality of tile headers and bitstreams. In bitstream BS0, codes are set in turn from LL as a subband corresponding to the lowest resolution in ascending order of resolution, as shown in Fig. 23A.

The image decoding unit 125 sequentially reads this bitstream, and displays an image upon completion of decoding of codes of each bit plane. Fig. 23B shows correspondence between respective subbands, and the sizes of images to be displayed. In this example, two levels of two-dimensional discrete wavelet transformation processes

Note that this embodiment adopts the format shown in Fig. 5D as that of the bitstream of each tile. This embodiment adopts such format since the respective segmented bitstreams have different resolutions. Alternatively, the respective segmented bitstreams may have different image qualities. In this case, the bitstream may have a format in which bit planes are arranged in the order from upper to lower bit planes, and each bit plane contains bitstreams of respective subbands, as shown in Fig. 25.

Also, when bitstreams for respective bit planes are formed in a bitstream of each tile, and bitstreams for respective subbands are formed in the bitstream of each bit plane, a hierarchical structure based on spatial positions of an image can be realized.

In addition, a hierarchical structure based on luminance components or color components of an image may be realized.

[Second Embodiment]

In the first embodiment, key generation is done in favor of tiles. That is, when a tile key is used in decryption, all resolution levels of that tile can be decrypted. However, the present invention is not limited to such specific process, and key generation may be done in favor of resolutions. That is, when a resolution key is used in decryption, all tiles of that resolution level

it is determined whether or not tile keys K_TjSi corresponding to all tiles of resolution i have been generated. If the checking result is true, the flow advances to step S148 to increment variable j by 1 (step S148), and the process in step S146 is executed. On the other hand, if the checking result in step S147 is false, the flow advances to step S149 to check if variable i < parameter N (step S149). Note that parameter N is the number of resolution levels. That is, it is determined whether or not resolution keys K_Si corresponding to all resolution levels have been generated. If the checking result is true, the flow advances to step S1410, and the processes in step S144 and subsequent steps are executed. On the other hand, if the checking result is false, the process ends.

A set of keys and key IDs generated by the aforementioned process will be described below using Fig. 24. Fig. 24 shows an example wherein an image is made up of four tiles, and each tile has three resolution levels.

In step S141, image key K_IMG and its key ID "0000" are generated. In step S142, bitstream key K_BS and its key ID "0001" are generated. In step S144, resolution key K_S1 and its key ID "0001", resolution key K_S2 and its key ID "0002", and resolution key K_S3 and its key ID "0003" are generated. In practice, since bitstream key K_BS is equal to resolution key K_S1 of the highest resolution, the

bitstreams in a code sequence may be encrypted using only one encryption key.

Note that the predetermined bitstreams mean those corresponding to predetermined tiles, those corresponding to predetermined subbands, predetermined bit planes (or a layer as a set of some bit planes), or the like. For example, by encrypting only bitstreams corresponding to predetermined tiles, decryption control based on the spatial positions of an image can be implemented. Also, by encrypting only bitstreams corresponding to predetermined subbands, decryption control based on the frequency components of an image can be implemented. Furthermore, by encrypting only predetermined bit planes (or a layer as a set of some bit planes), decryption control based on the image qualities of an image can be implemented.

When only predetermined bitstreams in a code sequence are encrypted using only one key, as described above, a plurality of keys need not be used, but decryption requires information indicating whether or not bitstreams in the code sequence have been encrypted. As such information, the ID reference table shown in Fig. 10 may be modified and used. In this embodiment, information indicating encrypted bitstreams in a code sequence will be referred to as an encryption map. The encryption map will be described below using Figs. 14A, 14B, and 14C.

Fig. 14A shows encrypted predetermined bitstreams (halftone dot portions) in a code sequence which has a start

since the code sequence has already been encrypted, an encryption process is skipped. On the other hand, if an encryption map is not included (false in step S161), the flow advances to step S162. In step S161, if the encryption map has been encrypted, a decryption process is executed. Furthermore, if the encryption map has been signed, a signature verification process is executed. For example, even when the encryption map is found, if a signature is not verified normally, the process ends.

On the other hand, predetermined bitstreams to be encrypted in the code sequence are determined, and an encryption map is generated and saved based on the determined contents (step S162). Note that the predetermined bitstreams to be encrypted may be explicitly designated by the user or may be automatically selected. Then, the predetermined bitstreams in the code sequence are encrypted on the basis of the generated encryption map (step S163).

With the aforementioned process, the encryption map can be generated while preventing double encryption, and an encryption process can be executed.

The decryption process of this embodiment will be described below. The decryption process of this embodiment is executed by the decryption apparatus with the arrangement shown in Fig. 11. In this embodiment, however, the first key generation unit 121, network unit 122, and second key generation unit 123 are not always required, and

However, if the decryption process is not normally terminated due to, e.g., use of a wrong key or the like, the record of this bitstream may be left unerased.

It is then checked if variable i < variable A (step S175). Note that variable A is the total number of encrypted bitstreams in the code sequence. For example, in the example shown in Fig. 14A, A = "3". That is, it is determined in step S175 whether or not all bitstreams have been decrypted. If the checking result is true, the flow advances to step S176 to increment variable i by 1, and the processes in step S173 and subsequent steps are executed. On the other hand, if the checking result is false, the process ends.

With the above process, the decryption process can be executed while preventing double decryption.

Note that the encryption map may be compressed to allow its efficient recording or transmission. Furthermore, when the encryption map of this embodiment is used, image decoding processes of some encrypted bitstreams may be skipped using the method described in the second embodiment.

In the encryption process, various bitstreams (e.g., bitstreams in respective layer levels) in a code sequence may be encrypted using a plurality of different keys as in the first and second embodiments.

[Fourth Embodiment]

In this embodiment, the encryption map must be securely transmitted from the encryption apparatus to the decryption apparatus. For this purpose, the encryption apparatus may encrypt the encryption map, and the decryption apparatus may decrypt it. Alternatively, the encryption apparatus may encrypt the encryption map, and the decryption apparatus may verify it.

Note that the encryption map may be compressed to allow its efficient recording or transmission. Furthermore, when the encryption map of this embodiment is used, image decoding processes of some encrypted bitstreams may be skipped using the method described in the second embodiment.

[Fifth Embodiment]

In the above embodiments, the decryption process is executed with reference to the ID reference table (or encryption map). In this case, a record corresponding to a portion that has undergone the decryption process may be erased from the ID reference table (or encryption map). By erasing records, it is determined by checking only the ID reference table (or encryption map) that portions which have records in the ID reference table (or encryption map) are encrypted, while portions which do not have any record in the ID reference table (or encryption map) are not encrypted.

Note that such process can be executed only when the ID reference table (or encryption map) is securely

signal having low-frequency components. This may pose a problem when the use wants to "clearly browse a low-resolution image (without any noise)". Hence, the aforementioned technique that skips an image decoding process of an encrypted bitstream is effective in such case.

Fig. 18 is a block diagram showing the functional arrangement of a decryption apparatus (decryption system) according to this embodiment. The decryption apparatus (decryption system) shown in Fig. 18 comprises a first key generation unit 191, network unit 192, second key generation unit 193, code sequence decryption unit 194, control unit 195, and image decoding unit 196. Note that the network unit 192 may be omitted. In this case, the decryption apparatus comprises the first key generation unit 191, second key generation unit 193, code sequence decryption unit 194, control unit 195, and image decoding unit 196. Note that the first key generation unit 191, network unit 192, second key generation unit 193, code sequence decryption unit 194, and image decoding unit 196 in Fig. 18 make the same operations as those of the first key generation unit 121, network unit 122, second key generation unit 123, code sequence decryption unit 124, and image decoding unit 125, and a description thereof will be omitted. The control unit 195 will be described below.

Fig. 19 is a block diagram showing the functional arrangement of the control unit 195 according to this

control unit 201, and outputs only non-encrypted bitstreams or packets in the entire code sequence in accordance with the information associated with bitstreams or packets.

As described above, whether or not each bitstream or packet is encrypted is determined using the ID reference table or encryption map, and only non-encrypted bitstreams or packets are decoded. Hence, the user can browse a permitted image free from noise.

[Another Embodiment]

The objects of the present invention are also achieved by supplying a recording medium (or storage medium), which records a program code of a software program that can implement the functions of the above-mentioned embodiments to the system or apparatus, and reading out and executing the program code stored in the recording medium by a computer (or a CPU or MPU) of the system or apparatus. In this case, the program code itself read out from the recording medium implements the functions of the above-mentioned embodiments, and the recording medium which stores the program code constitutes the present invention.

The functions of the above-mentioned embodiments may be implemented not only by executing the readout program code by the computer but also by some or all of actual processing operations executed by an operating system (OS) running on the computer on the basis of an instruction of the program code.

CLAIMS

1. An information processing method characterized by comprising:

a first recognition step of recognizing a first unit n (n = 1 to N) which specifies segmentation of encoded image data;

a second recognition step of recognizing a second unit m (m = 1 to M) which specifies segmentation of encoded image data;

a first parameter generation step of generating a first parameter X_n (n = 1 to N) on the basis of the first unit n; and

a key parameter generation step of generating a key parameter corresponding to each combination (n, m) of the first and second units on the basis of at least the first parameter, and

in that the key parameter is used to encrypt or decrypt partial encoded image data D(n, m) corresponding to the combination (n, m) in the encoded image data, and

the first parameter generation step includes a step of generating each first parameter X_n on the basis of an algorithm which uniquely determines the first parameter X_n based on a neighboring first parameter X_{n-1}.

2. The method according to claim 1, characterized in that an image expressed by the encoded image data is made

9. An information processing apparatus characterized by comprising:

first recognition means for recognizing a first unit n (n = 1 to N) which specifies segmentation of encoded image data;

second recognition means for recognizing a second unit m (m = 1 to M) which specifies segmentation of encoded image data;

first parameter generation means for generating a first parameter X_n (n = 1 to N) on the basis of the first unit n; and

key parameter generation means for generating a key parameter corresponding to each combination (n, m) of the first and second units on the basis of at least the first parameter, and

in that the key parameter is used to encrypt or decrypt partial encoded image data D(n, m) corresponding to the combination (n, m) in the encoded image data, and

said first parameter generation means generates each first parameter X_n on the basis of an algorithm which uniquely determines the first parameter X_n based on a neighboring first parameter X_{n-1}.

10. A program for making a computer execute an information processing method of claim 1.

FIG. 1

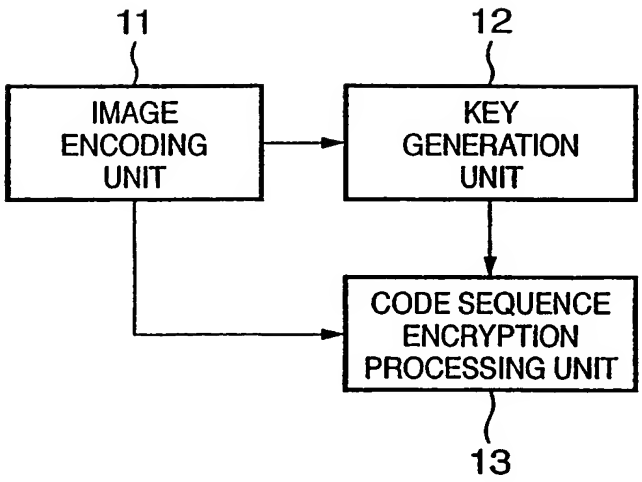

11
12
IMAGE ENCODING UNIT
KEY GENERATION UNIT
CODE SEQUENCE ENCRYPTION PROCESSING UNIT
13

FIG. 2

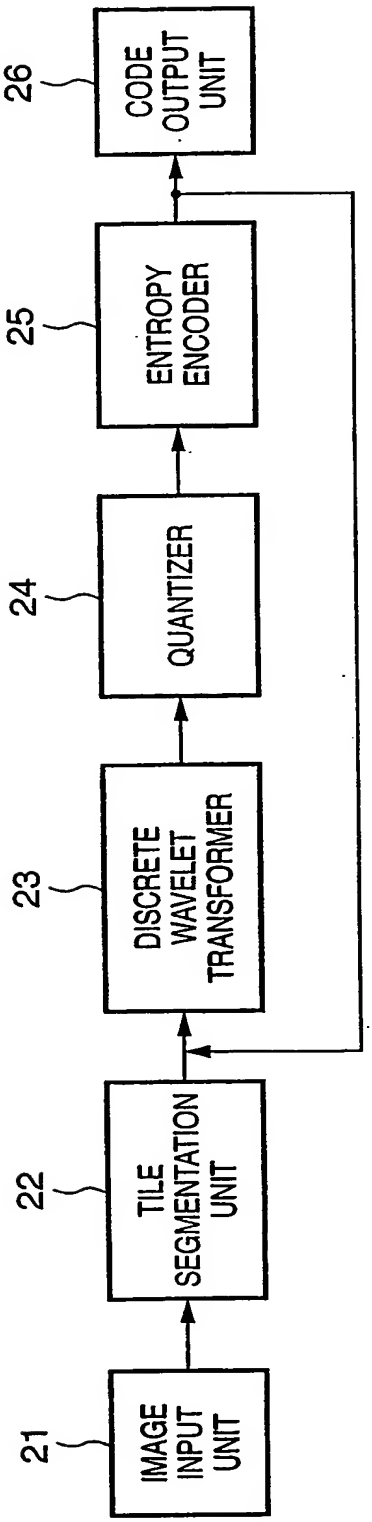

21
IMAGE INPUT UNIT
22
TILE SEGMENTATION UNIT
23
DISCRETE WAVELET TRANSFORMER
24
QUANTIZER
25
ENTROPY ENCODER
26
CODE OUTPUT UNIT

FIG. 3A

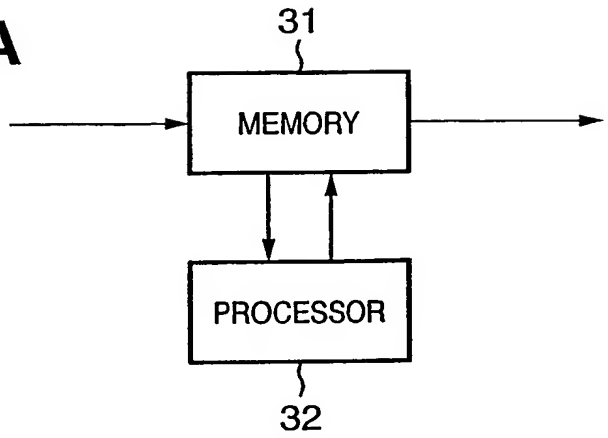


FIG. 3B

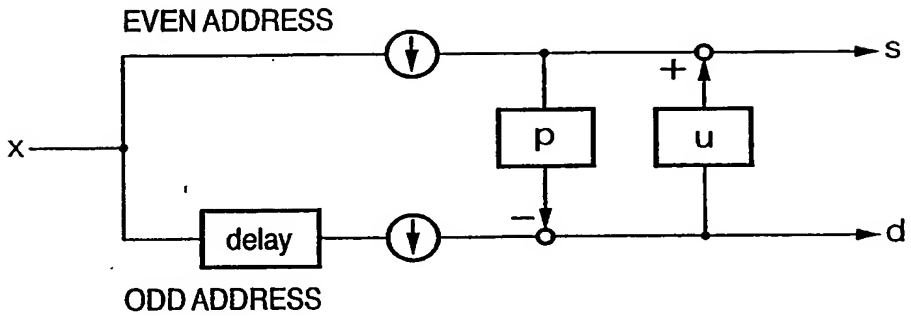


FIG. 3C

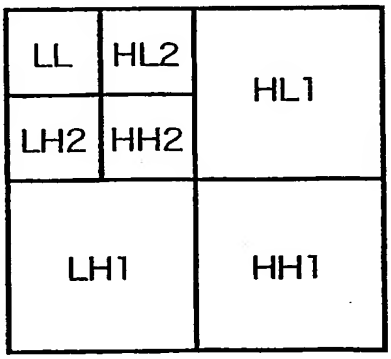


FIG. 4

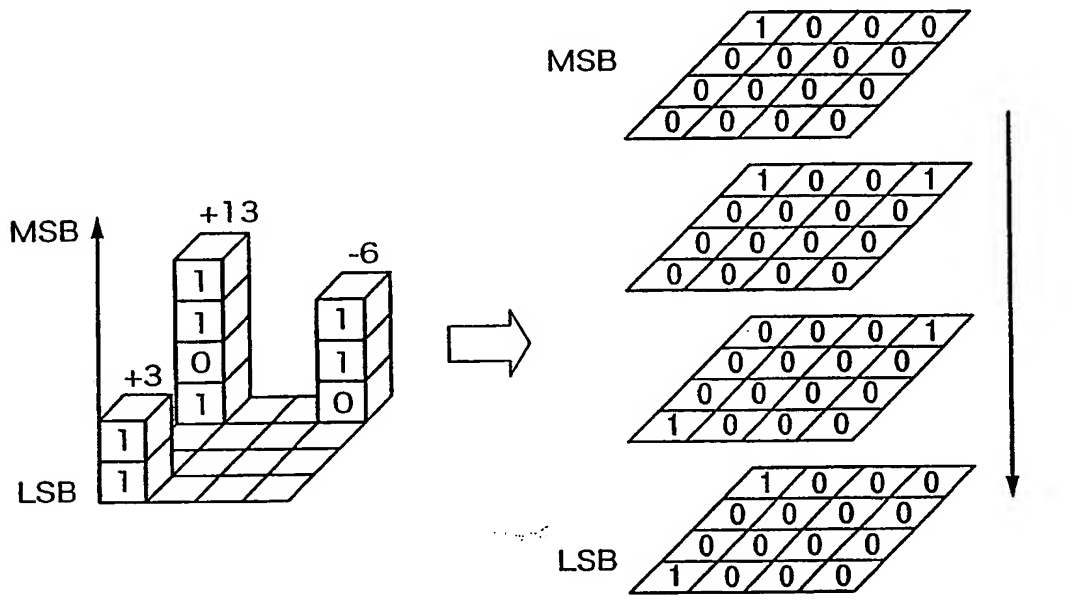


FIG. 5A

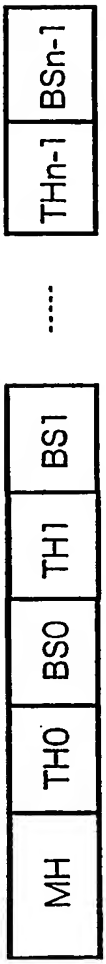


FIG. 5B

FIG. 5C

FIG. 5D

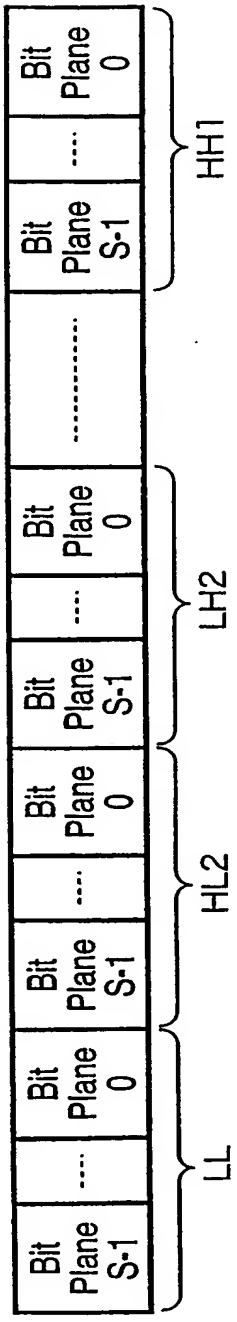


FIG. 6

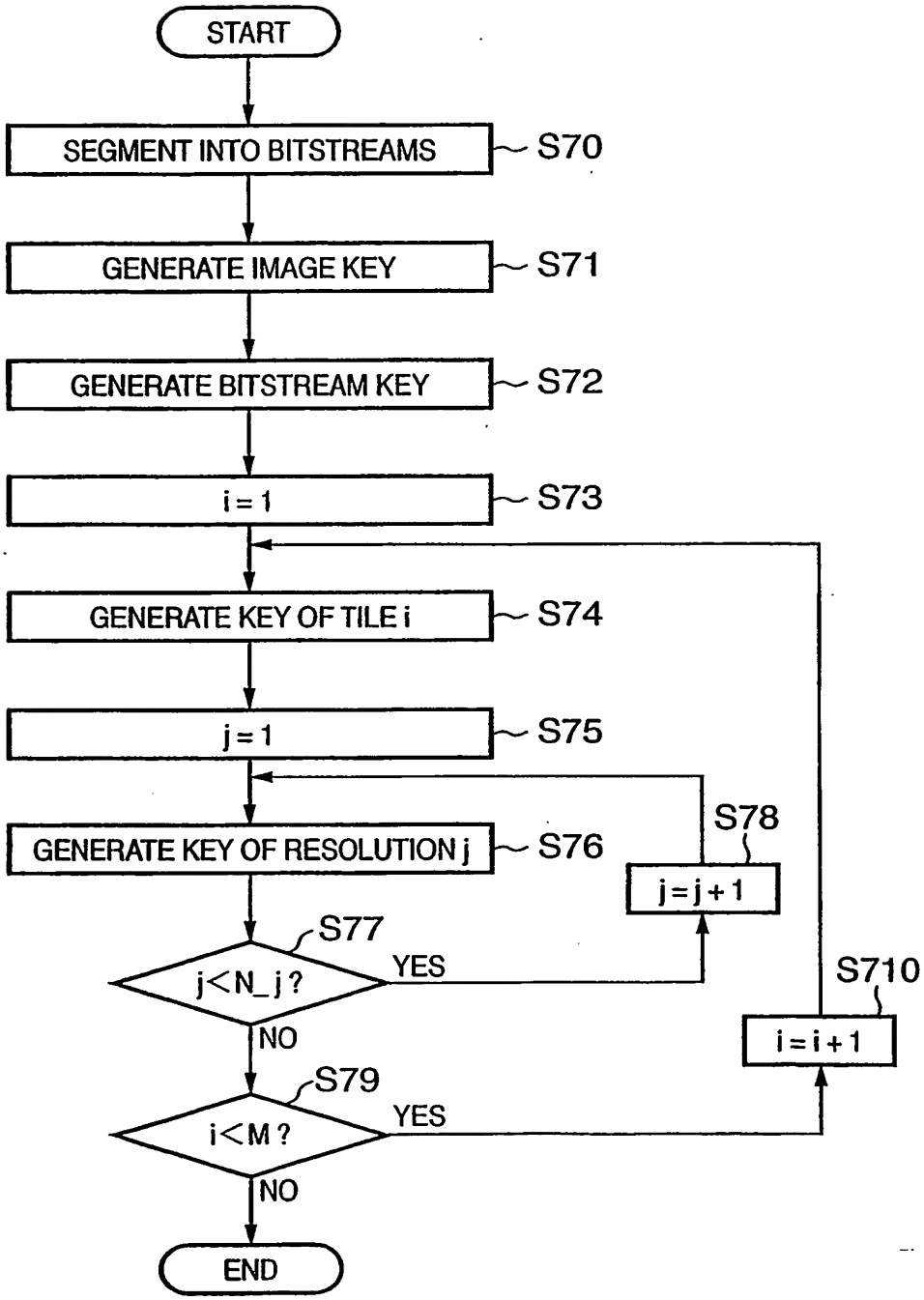

START
SEGMENT INTO BITSTREAMS
S70
GENERATE IMAGE KEY
S71
GENERATE BITSTREAM KEY
S72
i = 1
S73
GENERATE KEY OF TILE i
S74
j = 1
S75
GENERATE KEY OF RESOLUTION j
S76
S78
j = j + 1
S77
j<N_j ?
YES
NO
S710
i = i + 1
S79
i<M ?
YES
NO
END

FIG. 7

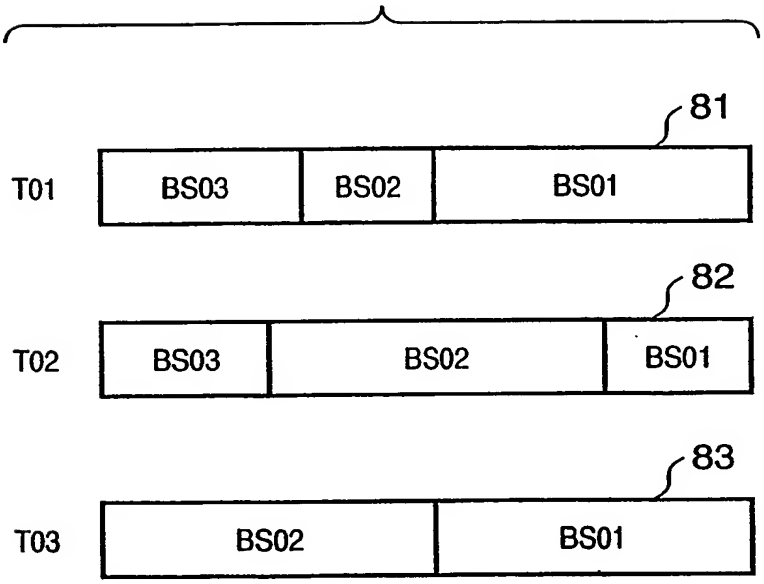


FIG. 8

TILE INDEX	RESOLUTION INDEX

FIG. 9

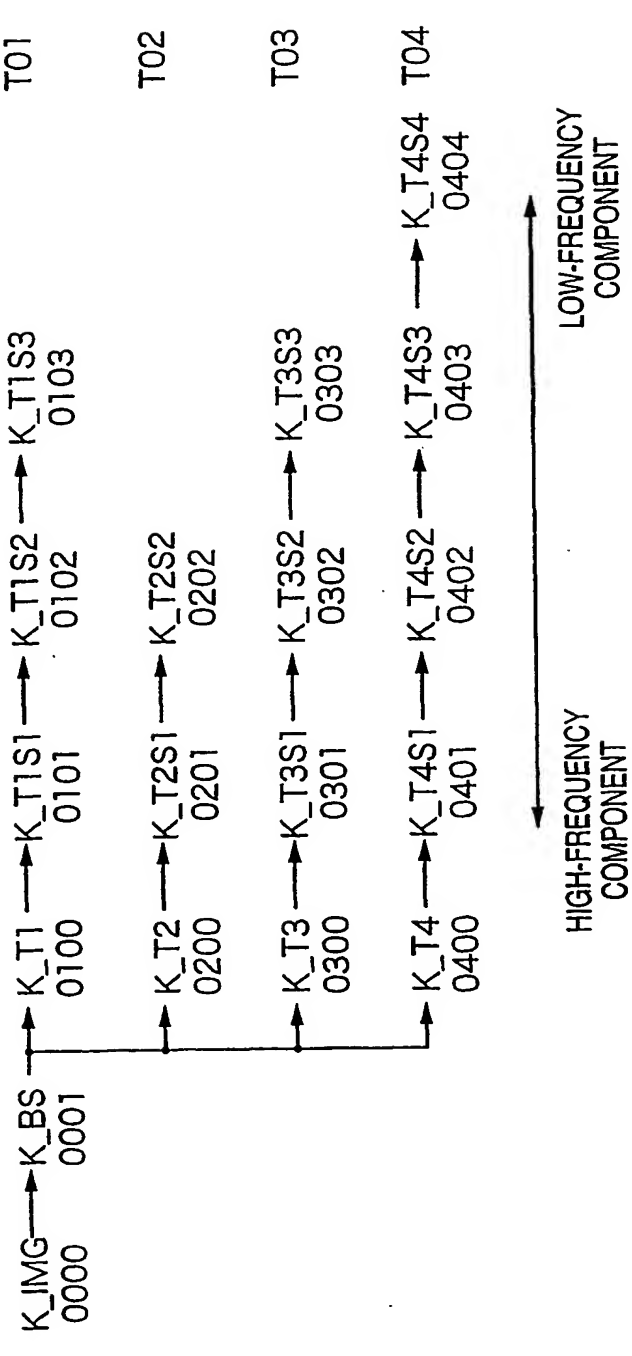

K_IMG
0000
K_BS
0001
K_T1
0100
K_T2
0200
K_T3
0300
K_T4
0400
K_T1S1
0101
K_T2S1
0201
K_T3S1
0301
K_T4S1
0401
K_T1S2
0102
K_T2S2
0202
K_T3S2
0302
K_T4S2
0402
K_T1S3
0103
K_T3S3
0303
K_T4S3
0403
K_T4S4
0404
T01
T02
T03
T04
LOW-FREQUENCY COMPONENT
HIGH-FREQUENCY COMPONENT

FIG. 10

ID	TILE INFORMATION	OFFSET	LENGTH
0101	(0,0)	0	7
0102	(0,0)	(9)	8
0103	(0,0)	(15)	5
:	:	:	:
0202	(64,0)	(12)	9
:	:	:	:

FIG. 11

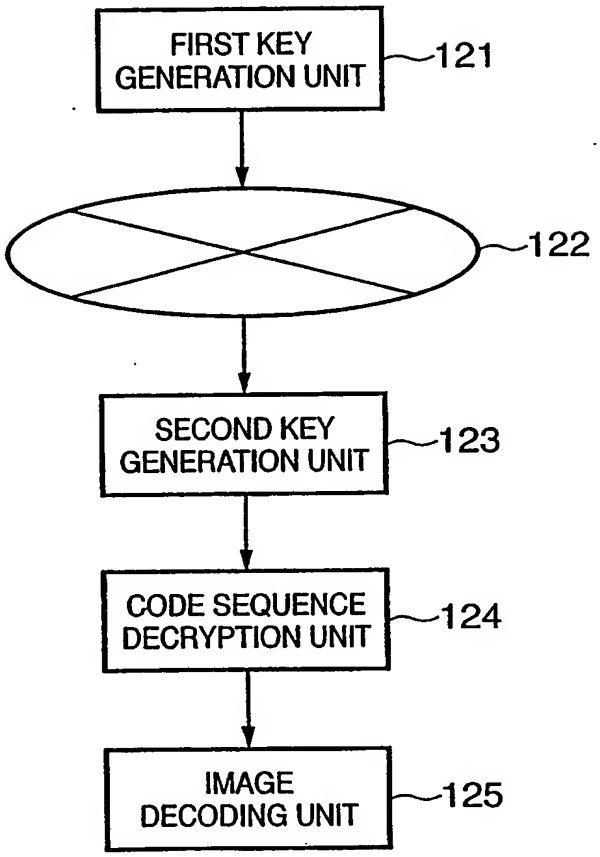

FIRST KEY GENERATION UNIT
121
122
SECOND KEY GENERATION UNIT
123
CODE SEQUENCE DECRYPTION UNIT
124
IMAGE DECODING UNIT
125

FIG. 12

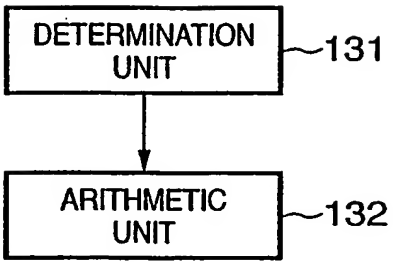


FIG. 13

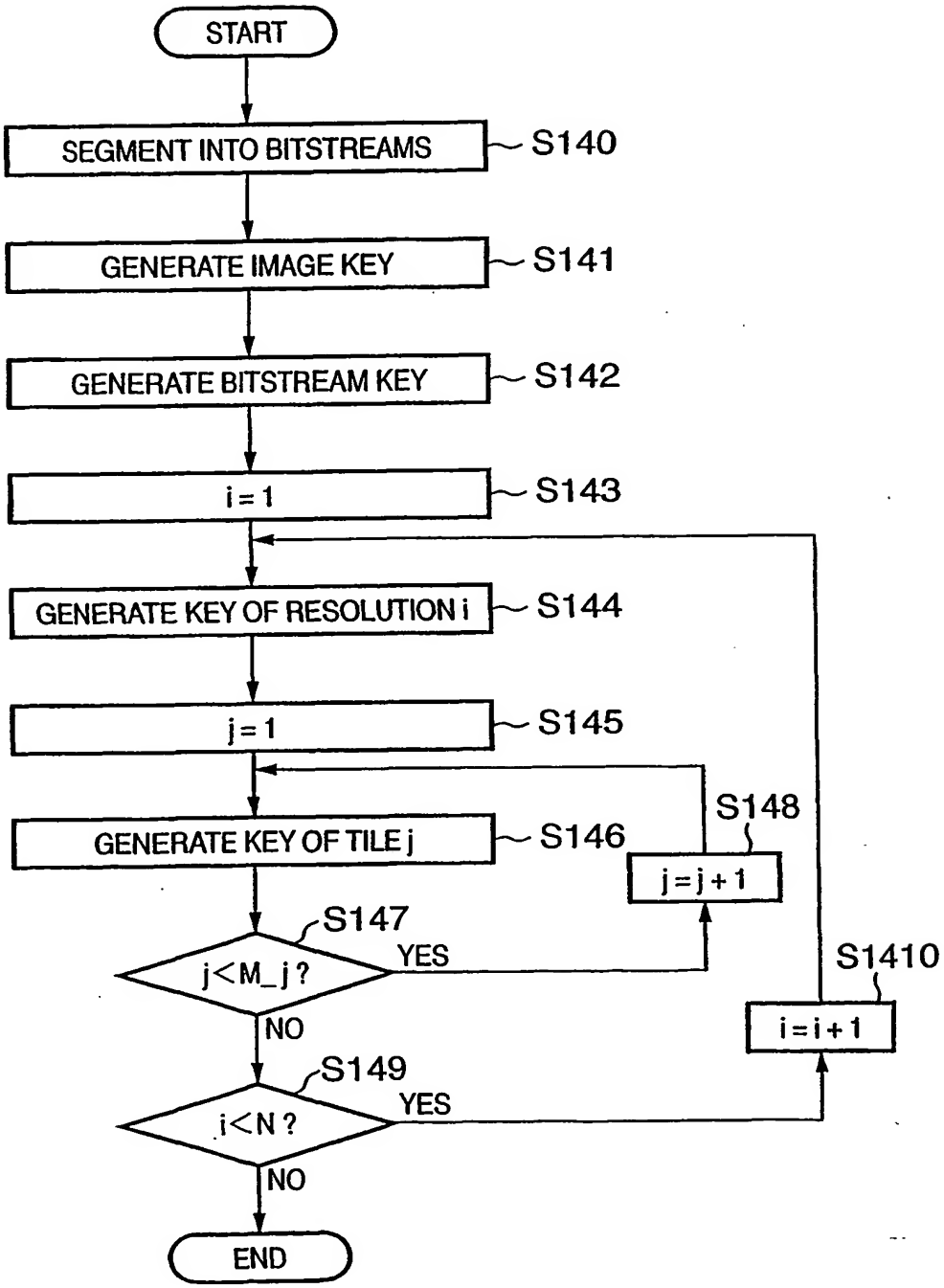

START
SEGMENT INTO BITSTREAMS
S140
GENERATE IMAGE KEY
S141
GENERATE BITSTREAM KEY
S142
i=1
S143
GENERATE KEY OF RESOLUTION i
S144
j=1
S145
GENERATE KEY OF TILE j
S146
S148
j=j+1
S147
j<M_j?
YES
NO
S1410
i=i+1
S149
i<N?
YES
NO
END

FIG. 14A

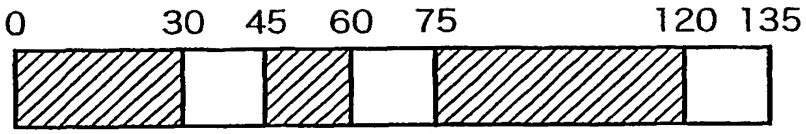


FIG. 14B

OFFSET	LENGTH
0	30
45	15
75	45

FIG. 14C

START POINT	END POINT
0	30
45	60
75	120

FIG. 15

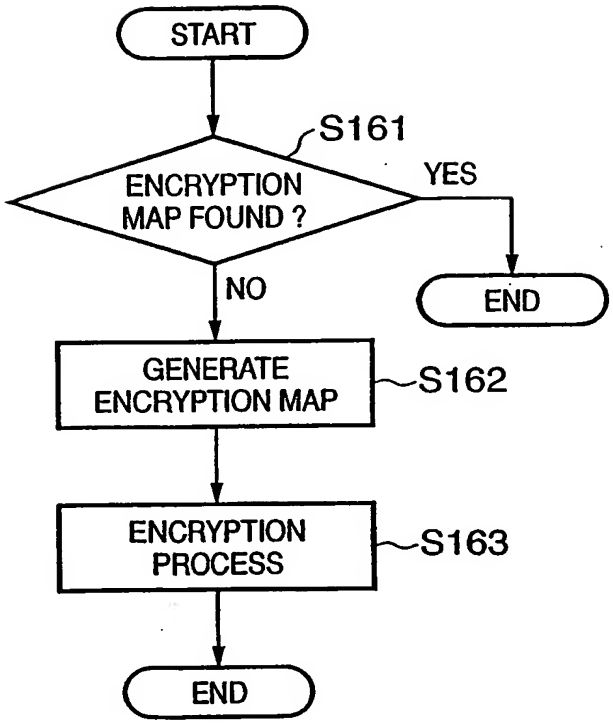

START
S161
ENCRYPTION MAP FOUND ?
YES
END
NO
GENERATE ENCRYPTION MAP
S162
ENCRYPTION PROCESS
S163
END

FIG. 16

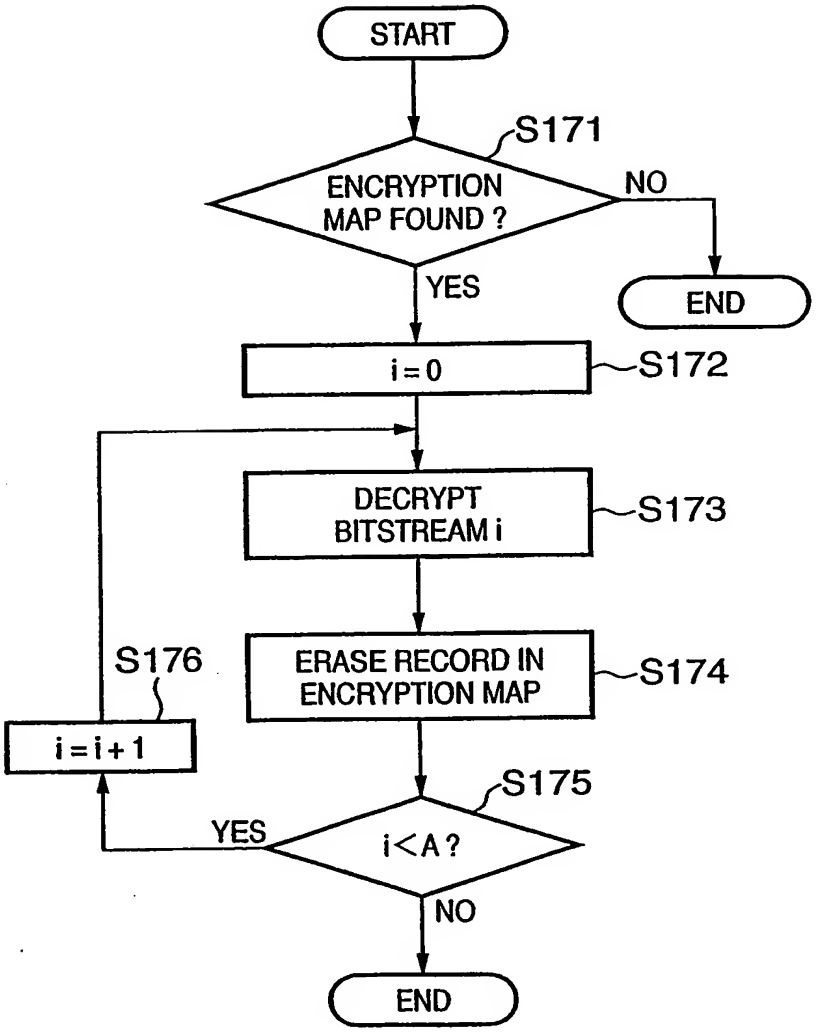

START
S171
ENCRYPTION MAP FOUND ?
NO
END
YES
i=0
S172
DECRYPT BITSTREAM i
S173
ERASE RECORD IN ENCRYPTION MAP
S174
S176
i=i+1
S175
YES
i<A ?
NO
END

FIG. 17A

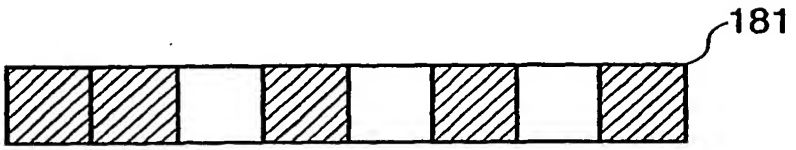


FIG. 17B

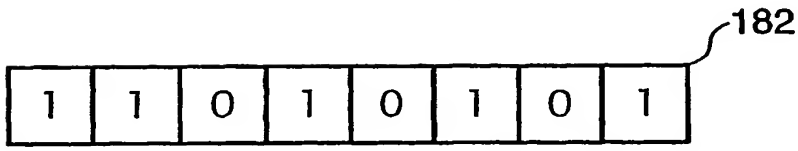


FIG. 18

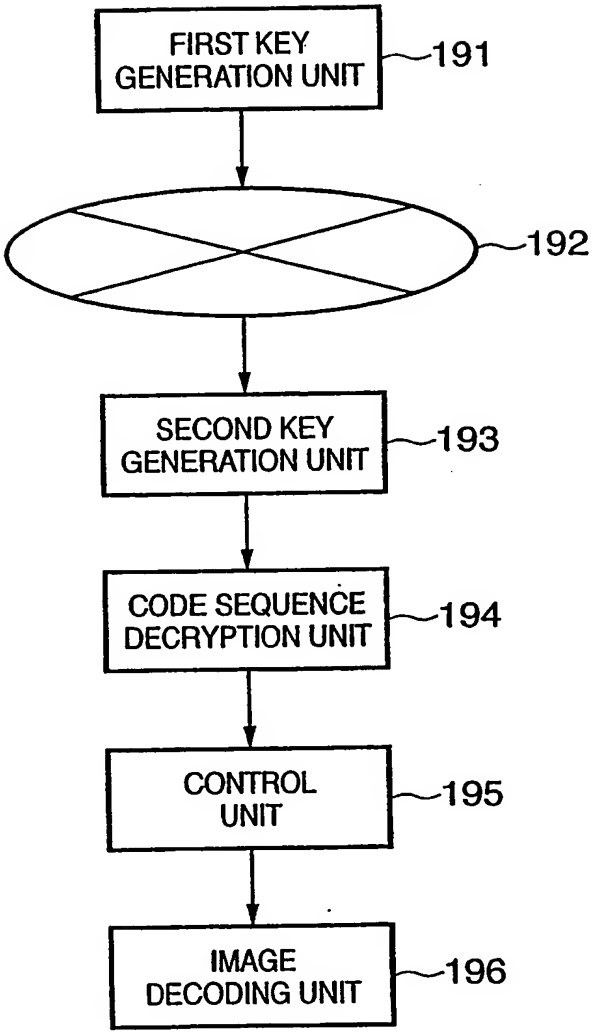

FIRST KEY GENERATION UNIT
191
192
SECOND KEY GENERATION UNIT
193
CODE SEQUENCE DECRYPTION UNIT
194
CONTROL UNIT
195
IMAGE DECODING UNIT
196

FIG. 19

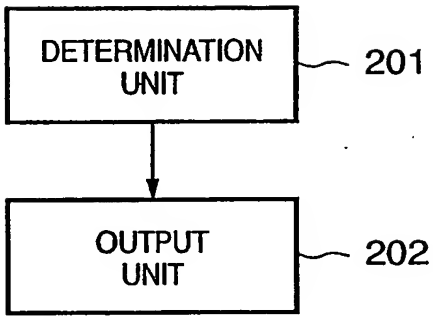


FIG. 20

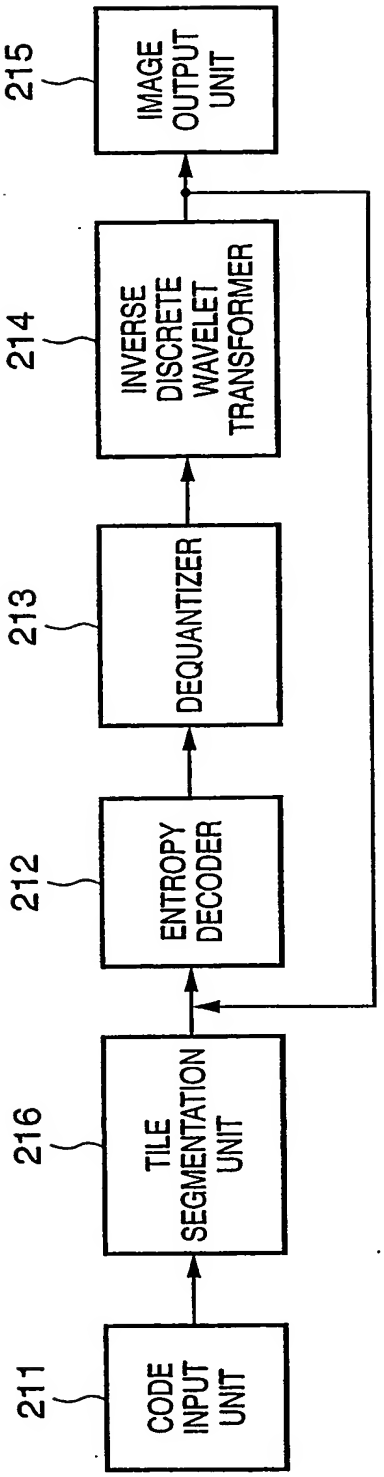

211
CODE INPUT UNIT
216
TILE SEGMENTATION UNIT
212
ENTROPY DECODER
213
DEQUANTIZER
214
INVERSE DISCRETE WAVELET TRANSFORMER
215
IMAGE OUTPUT UNIT

FIG. 21

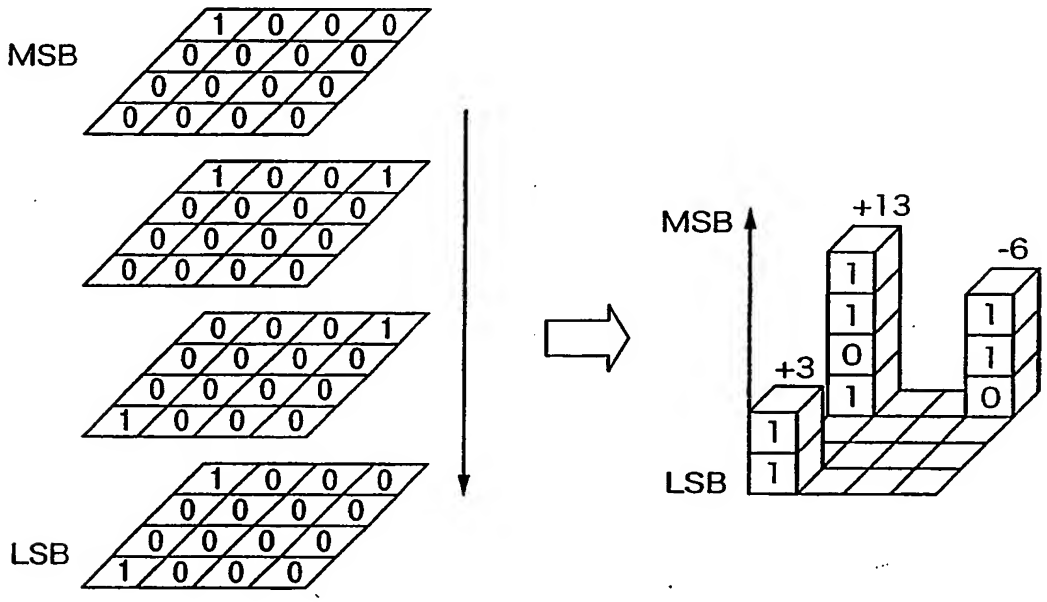


FIG. 22A

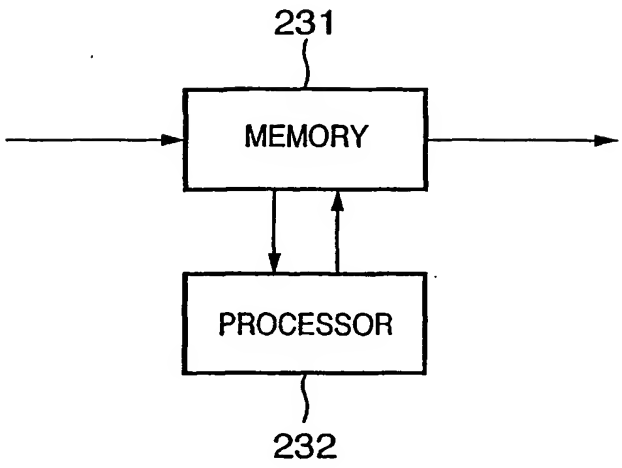


FIG. 22B

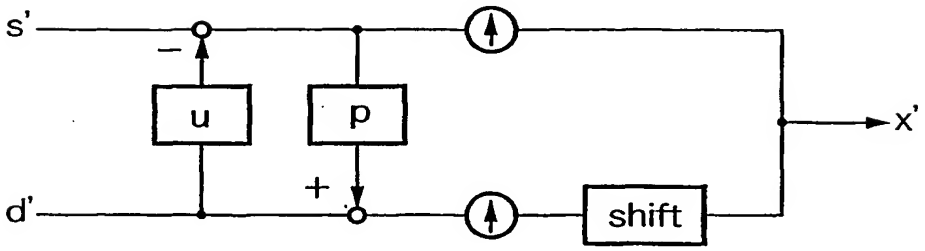


FIG. 23A

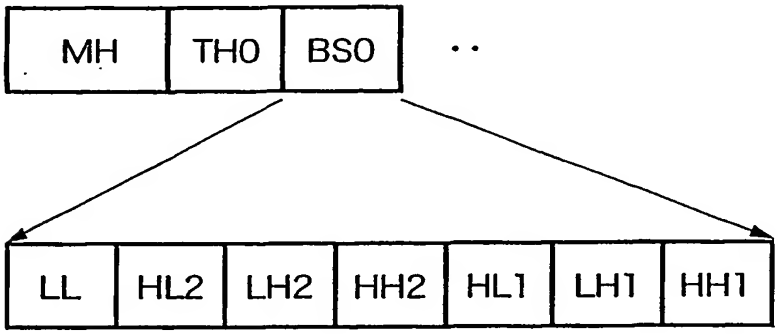
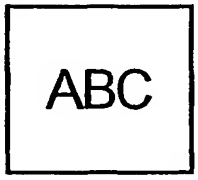
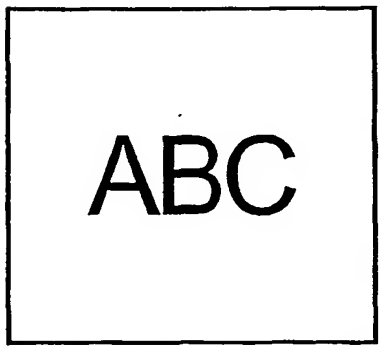


FIG. 23B

=LL



=LL+HL2+LH2+HH2



=LL+HL2+LH2+HH2+HL1+LH1+HH1

FIG. 24

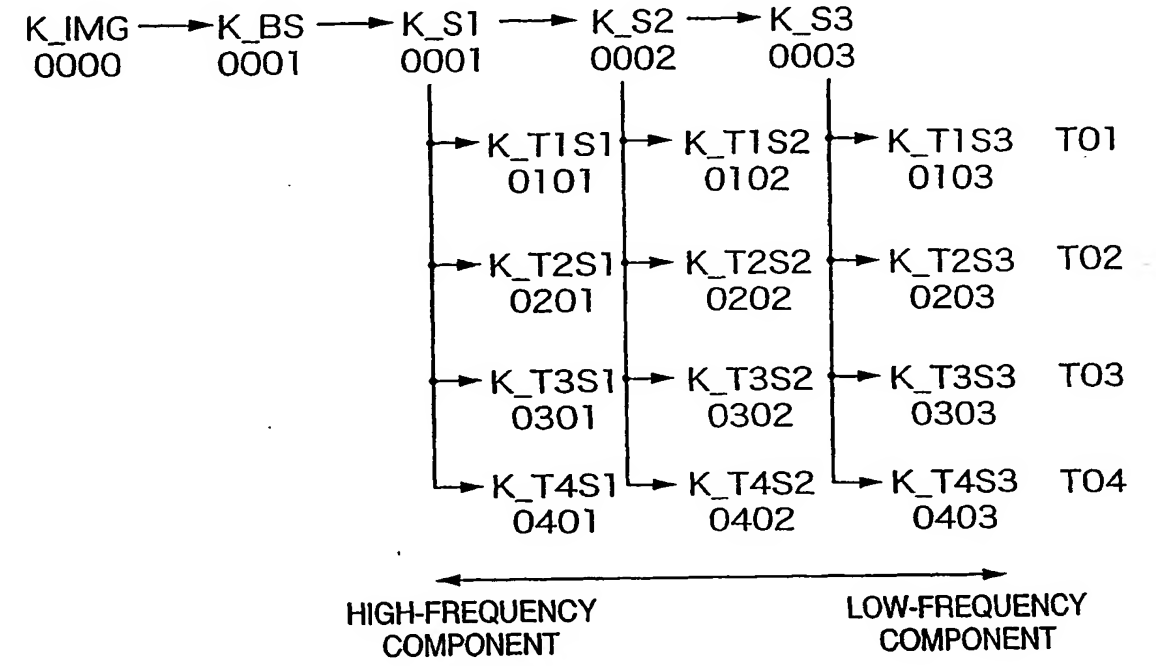
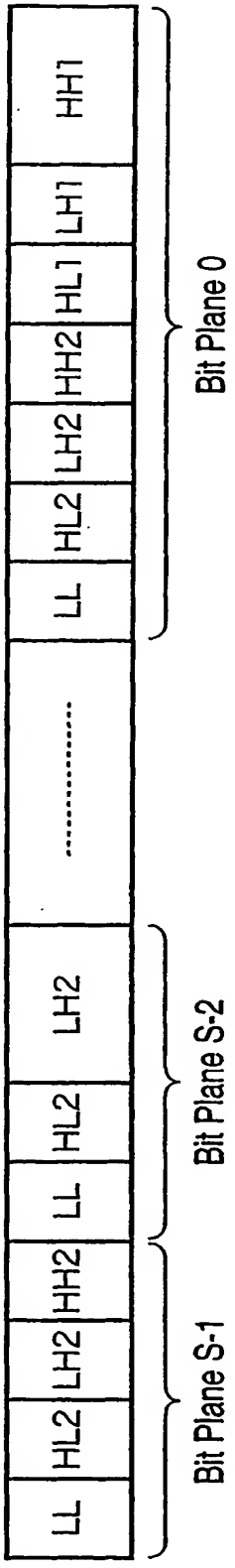

K_IMG
0000
K_BS
0001
K_S1
0001
K_S2
0002
K_S3
0003
K_T1S1
0101
K_T1S2
0102
K_T1S3
0103
T01
K_T2S1
0201
K_T2S2
0202
K_T2S3
0203
T02
K_T3S1
0301
K_T3S2
0302
K_T3S3
0303
T03
K_T4S1
0401
K_T4S2
0402
K_T4S3
0403
T04
HIGH-FREQUENCY COMPONENT
LOW-FREQUENCY COMPONENT

FIG. 25


LL
HL2
LH2
HH2
LL
HL2
LH2
.....
LL
HL2
LH2
HH2
HL1
LH1
HH1
Bit Plane S-1
Bit Plane S-2
Bit Plane 0

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/07976

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl[7] H04L9/16 H04L9/08 H04N1/41 H04N1/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl[7] H04L9/16 H04L9/08 H04N1/41 H04N1/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Japanese Utility Model Gazette 1926-1996, Japanese Publication of Unexamined Utility Model Applications 1971-2003, Japanese Registered Utility Model Gazette 1994-2003, Japanese Gazette Containing the Utility Model 1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 11-18070 A(MATSUSHITA DENKI SANGYOU K.K.) 1999.01.22, All pages and Figure 1-15 (Family:None)	1-11
Y	JP 9-327010 A(NIPPON DENSHIN DENWA K.K.) 1997.12.16, No.[0033]-[0037]paragraph, Figure2 (Family:None)	1-11
Y	JP 7-15715 A(NIPPON DENKI K.K.) 1995.01.17, All pages and Figure 1-2 (Family:None)	1-11

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17.09.03	30.09.03

Name and mailing address of the ISA/JP	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Shigenori AOKI 5M 4229 Telephone No. +81-3-3581-1101 Ext. 3597

Form PCT/ISA/210 (second sheet) (July 1998)